



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/800,056	03/12/2004	Matthew Labarge	50037.218US01	8450
27488	7590	12/28/2007		
MERCHANT & GOULD (MICROSOFT)			EXAMINER	
P.O. BOX 2903			NGUYEN, CHAU T	
MINNEAPOLIS, MN 55402-0903			ART UNIT	PAPER NUMBER
			2176	
			MAIL DATE	DELIVERY MODE
			12/28/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/800,056

Applicant(s)

LABARGE ET AL.

Examiner

Chau Nguyen

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 October 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-25 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>10/29/2007</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Applicant's amendment filed on 10/15/2007 has been entered. Claims 1-25 are pending. Claims 1, 11 and 20 are independent claims.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1 and 3-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burkett et al. (Burkett), US Patent No. 6,635,089 in view of Pik et al. (Pik), US Patent Application Publication No. US 2004/0230906, and further in view of Messina, US Patent No. 5,634,128.

4. As to independent claims 1, 11 and 20, Burkett discloses computer-implemented method for binding data in a user interface (UI) script, comprising:

generating a tree structure that corresponds to the UI script (Figs. 4A&4B and Figs 5A-5E, col. 9, line 44 – col. 11, line 14: the DOM tree (tree structure) is generated and the DOM tree corresponds to the XML document (UI script)); wherein generating the tree structure includes automatically determining templates that were previously

Art Unit: 2176

grafted to the tree and automatically removing any template that were previously grafted to the tree (Burkett discloses the DOM tree (tree structure) is generated (col. 9, line 11- col. 11, line 14). Burkett also discloses that the DOM API enables application program (automatically) to access a tree-oriented abstraction of a document, and to manipulate document structure and contents (that is, by changing, deleting, and/or adding elements), i.e., when a query-results node is present, the parsed query result is inserted into the DOM tree as a subtree beneath that node, replacing any subtree that previously existed (col. 18, lines 34-38). Further, the DOM enables navigating the structure of the document (col. 1, lines 35-58). Thus, Burkett discloses generating the tree structure includes automatically determining templates that were previously grafted to the tree and automatically removing any templates that were previously grafted to the tree;

accessing a reference template (Abstract)

inserting the data into a portion of the tree structure and grafting the portion of the tree into the tree structure (Figs. 4A-4E and col. 9, line 44 – col. 12, line 4: dynamically constructed nodes 421a, 422a and 423a in Fig. 4E have been substituted for nodes 415, 126 and its subtree comprised of nodes 417, 418a, 418b, and 419);

the UI output is dynamically updated with the data (col. 3, lines 59-67)

However, Burkett does not explicitly disclose cloning the reference template to create a cloned reference template while maintaining the reference template, inserting the data into the cloned reference template and grafting the cloned reference template into the tree structure after the data has been inserted into the cloned reference template and displaying a UI output according to the tree structure.

Since Burkett discloses substitute one portion of the tree with different data (Figs. 4A-4E and col. 9, line 44 – col. 12, line 4: dynamically constructed nodes 421a, 422a and 423a in Fig. 4E have been substituted for nodes 415, 126 and its subtree comprised of nodes 417, 418a, 418b, and 419), thus, the step of cloning a portion of the tree structure should be inherent in the above substitute step. In addition, Pik discloses storing the scripting object can be involve copying a scripting object reference to the shared memory to cause retention of the scripting object by the client during document loading, then cloning the stored scripting object to the newly loaded document can be performed and can involve creating a new scripting object in the new document and copying data from the stored scripting object to the new scripting object (Abstract, Figures 2-3). Pik also discloses in pages 3-4, paragraphs [0028]-[0035] that the events are user triggered events that change the state of a scripting object loaded in the page, and during the document reload, the stored scripting object can be cloned to the reloaded document, creating a cloned scripting object in a reloaded document memory space. Moreover, a reference to the scripting object of the loaded document can be copied to a shared memory and this causes retention of the scripting object during the document reload (page 3, paragraph [0032]). Figures 4-7 and page 4, paragraphs [0036]-[0039] of Pik reference provide a simple point and click user interface to a navigation hierarchy. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Pik and Burkett to include cloning the reference template to create a cloned reference template while maintaining the reference template, inserting the data into the cloned reference template and

grafting the cloned reference template into the tree structure after the data has been inserted into the cloned reference template and displaying a UI output according to the tree structure. Pik suggests that copying the reference to the object to the shared memory prevents the object from being released when the document is reloaded, thus saving the state information in the object. Also, storing the scripting object by copying a reference to it instead of copying the object itself, can result in better performance and an improved portal system (Pik, page 3, paragraph [0032]).

However, Burkett and Pik do not explicitly disclose automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source (bound data or a separate data source) are removed from the tree.

Messina discloses controlling the access to objects stored in a data processing system wherein the objects are hierarchically structured (tree dom) and each object being either locally available or retrievable from an external unit (col. 2, lines 14-25). Messina further discloses all of the objects contain information about the existence of dependent objects (a counter is associated with each object indicating whether there are objects dependent on the object) (col. 4, lines 1-6). Messina further discloses that through the dependent object counter, a processing program can determine if any locally available object has a dependent object, then appending a dependent object to a locally available object when at least one related dependent object is non-locally available (col. 4, lines 10-15). Messina further discloses that a place-holder graphic symbol (template) is displayed to represent all of the non-locally available dependent

Art Unit: 2176

objects, and when all of the non-locally available objects are retrieved, the place-holder graphic symbol (template) is removed from the display (tree) and the dependent object is deleted from the tree in the memory (col. 2, lines 34-48). Thus, these steps imply automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Messina with Burkett and Pik to include automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree for the purpose of improving the performance of applications involving the retrieval of non-locally available objects from external units.

5. As to dependent claims 3, 13 and 21, Burkett-Pik-Messina disclose wherein cloning a portion of the tree structure further comprises determining which portions of the tree structure correspond to a specified tag of the UI script (Burkett, col. 9, lines 44-45).

6. As to dependent claims 4, 14 and 22, Burkett-Pik-Messina disclose wherein the specified tag has an associated attribute for retrieving multiple records for display in the UI output as a list of records (Burkett, col. 11, lines 36 – col. 12, line 34).

Art Unit: 2176

7. As to dependent claims 5, 15 and 23, Burkett-Pik-Messina disclose wherein the specified tag has an associated attribute that specifies an interval for refreshing the data (Burkett, col. 13, lines 47-65).

8. As to dependent claims 6 and 16, Burkett-Pik-Messina disclose wherein inserting the data further comprises retrieving the data from an external data source (Burkett, col. 13, lines 6-20).

9. As to dependent claim 7, Burkett-Pik-Messina disclose wherein retrieving the data further comprises determining a location of the data according to a uniform resource locator (URL) within the UI script (Burkett, col. 13, lines 6-20).

10. As to dependent claims 8 and 17, Burkett-Pik-Messina disclose wherein retrieving the data further comprises passing a uniform resource locator (URL) that identifies a location of the data to a communication library (Burkett, col. 13, lines 6-20).

11. As to dependent claims 9, 18 and 24, Burkett-Pik-Messina disclose wherein a state is associated with each portion of the tree structure in which data is inserted (Burkett, col. 17, line 27 – col. 18, line 38).

12. As to dependent claims 10, 19 and 25, Burkett-Pik-Messina disclose wherein a first component is displayed in the UI when the state corresponds to a first state, and a

second component is displayed in the UI when the state corresponds to a second state (Pik, page 4, paragraphs [0036]-[0039]: Fig. 5 shows a reference to the scripting object is copied to the shared memory in the topmost frame, the page can then be reloaded and as shown in Figure 6, thus when only part of a page is to be reloaded with new information from a server, and another part of the page is to retain its current client-side state information, a reload of the entire page can be performed while preserving the desired client-side state information.

13. As to dependent claim 12, Burkett-Pik-Messina disclose wherein the tree structure and the UI script are logically equivalent (Burkett, col. 11, lines 1-14).

14. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Burkett et al. (Burkett), US Patent No. 6,635,089 and Pik et al. (Pik), US Patent Application Publication No. US 2004/0230906, and Messina, US Patent No. 5,634,128 as discussed in claims 1, 3-25 above, and further in view of Boehme et al. (Boehme), US Patent No. 6,578,192.

15. As to dependent claim 2, Burkett-Pik-Messina, however, do not explicitly disclose wherein the reference template includes a reference tag used to delineate components to which data binding is applied.

Boehme discloses a document is loaded having script components which are identified and delineated and then passed to an interpreter to returns an object

Art Unit: 2176

corresponding to each script component (Abstract and col. 3, line 59 – col. 4, line 38, and col. 7, lines 11-56).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Boehme with Burkett-Pik-Messina to include the reference template includes a reference tag used to delineate components to which data binding is applied for the purpose of generating object or desired content corresponding to each script component.

Response to Arguments

16. In the remarks, Applicant(s) argued in substance that

A) Burkett does not teach the use of a DOM tree that has been previously used (see page 7 of remarks).

In reply to argument A, Burkett discloses that the DOM API enables application program (automatically) to access a tree-oriented abstraction of a document, and to manipulate document structure and contents (that is, by changing, deleting, and/or adding elements), i.e., when a query-results node is present, the parsed query result is inserted into the DOM tree as a subtree beneath that node, replacing any subtree that previously existed or used (col. 18, lines 34-38).

B) Burkett does not teach the steps necessary to clean a reused tree up to avoid such errors (see page 7 of the remarks)

In reply to argument B, applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the steps necessary to clean a reused tree up to avoid such errors) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In this case, claim 1 claims "automatically removing any templates that were previously grafted to the tree". Burkett discloses that when a query-results node is present, the parsed query result is inserted into the DOM tree as a subtree beneath that node, replacing any subtree that previously existed or used (col. 18, lines 34-38). Thus, in order to inserting the parsed query result into the DOM to replace any subtree that previously existed or used, the subtree must be deleted or removed before the inserting and/or replacing steps.

C) Burkett does not teach generating a tree structure that corresponds to the UI script (see page 8 of remarks).

In reply to argument C, Burkett discloses in Figs. 4A&4B and Figs 5A-5E, col. 9, line 44 – col. 11, line 14: the DOM tree (tree structure) is generated and the DOM tree corresponds to the XML document (UI script).

Art Unit: 2176

D) Prior art does not teach automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree (see page 8 of remarks).

In reply to argument D, applicant's arguments with respect to newly amended limitation "such that templates that already include data from an external data source are removed from the tree" have been considered but are moot in view of the new ground(s) of rejection, i.e., in further view of Messina, US Patent No. 5,634,128.

17. Applicant's arguments filed 10/15/2007 have been fully considered but they are not persuasive. Please see the response to arguments above.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2176

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chau Nguyen whose telephone number is (571) 272-4092. The examiner can normally be reached on 8:30 am – 5:30 pm Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton, can be reached on (571) 272-4137. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. On July 15, 2005, the Central Facsimile (FAX) Number will change from 703-872-9306 to 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chau Nguyen
Patent Examiner
Art Unit 2176



DOUG HUTTON
SUPERVISORY PATENT EXAMINER